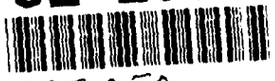
(2)

Measures of User-System Interface Effectiveness: Assessment of Structured Judgment Evaluation Techniques for Graphical, Direct-Manipulation Style Interfaces

MTR 92B0000047V2
July 1992

Donna L. Cuomo
Charles D. Bowen

MITRE
Bedford, Massachusetts



Measures of User-System Interface Effectiveness: Assessment of Structured Judgment Evaluation Techniques for Graphical, Direct-Manipulation Style Interfaces

MTR 92B0000047V2
July 1992

Donna L. Cuomo
Charles D. Bowen

Contract Sponsor N/A
Contract No. N/A
Project No. 91620
Dept. D047

MITRE
Bedford, Massachusetts

Department Approval: Nancy C. Goodwin

Nancy C. Goodwin

MITRE Project Approval: Donna L. Cuomo

Donna L. Cuomo

ABSTRACT

The results of the second phase of the MITRE sponsored research project on developing measures of user-system interface effectiveness are presented. This project is concerned with developing methods and measures of user-system interface effectiveness for command and control systems with graphical, direct manipulation style interfaces. Due to the increased use of user interface prototyping during concept definition and demonstration/validation phases, the opportunity exists for human factors engineers to apply evaluation methodologies early enough in the life cycle to make an impact on system design. Understanding and improving user-system interface (USI) evaluation techniques is critical to this process. In 1986, Norman proposed a descriptive "stages of user activity" model of human-computer interaction (HCI). Hutchins, Hollin, and Norman (1986) proposed concepts of measures based on the model which would assess the directness of the engagements between the user and the interface at each stage of the model. This phase of our research program involved applying three USI evaluation techniques to a single interface, and assessing which, if any, provided information on the directness of engagement at each stage of Norman's model. We also classified the problem types identified according to the Smith and Mosier (1986) functional areas. The three techniques used were cognitive walkthrough, heuristic evaluation, and guidelines. It was found that the cognitive walkthrough method applied almost exclusively to the action specification stage. Guidelines were applicable to more of the stages evaluated but all the techniques were weak in measuring semantic distance and all of the stages on the evaluation side of the HCI activity cycle. Improvements to existing or new techniques are required for evaluating the directness of engagement for graphical, direct-manipulation style interfaces.

EXECUTIVE SUMMARY

This paper discusses the results of the second phase of the MITRE sponsored research project on developing measures of user-system interface effectiveness. This project is concerned with developing methods and measures of user-system interface effectiveness for command and control systems with graphical, direct-manipulation style interfaces. Due to the increased use of user interface prototyping during concept definition and demonstration/validation phases, the opportunity exists for human factors engineers to apply evaluation methodologies early enough in the life cycle to make an impact on system design. Understanding and improving user-system interface (USI) evaluation techniques is critical to this process. We performed a study comparing three USI evaluation techniques to assess whether they provide adequate evaluation of graphical, direct-manipulation (DM) style interfaces. The types of problems identified by each method were classified according to two generic models of human-computer interaction (HCI). This part of the research was just one phase of an overall research program.

FORMAL FRAMEWORK OF HUMAN-COMPUTER INTERACTION (HCI)

Norman (1986) has proposed a descriptive model of human-computer interaction which describes a user's interaction with the computer as occurring in seven stages: establishing the goal, forming the intention, specifying the action sequence, executing the action, perceiving the system state, interpreting the system state, and evaluation of the system state with respect to the goals and intentions.

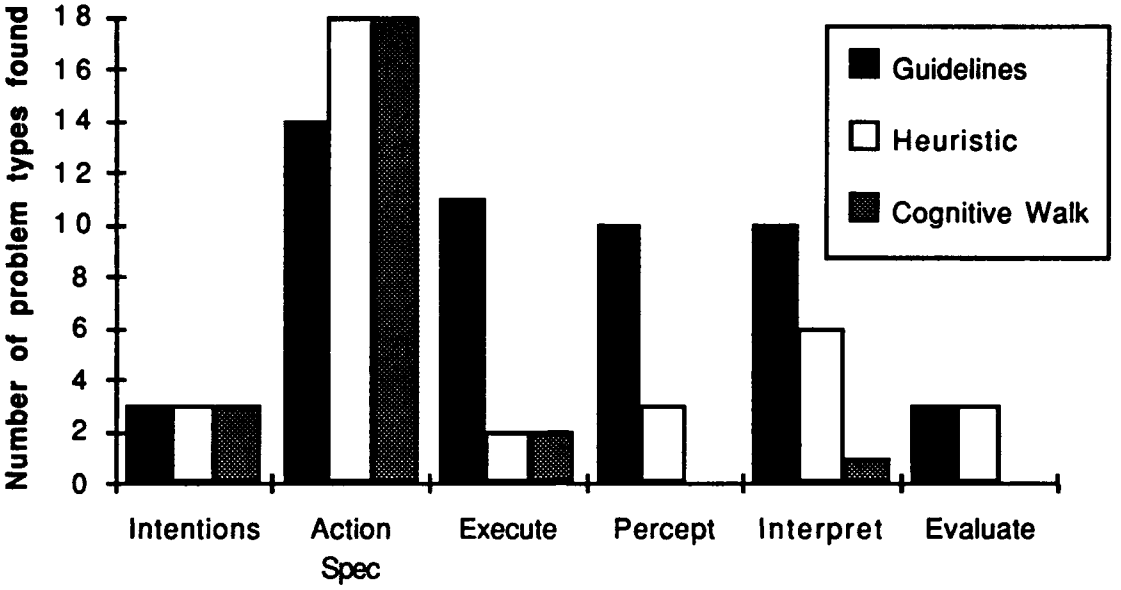
Hutchins, Hollin, and Norman (1986) proposed five concepts related to the directness of engagement of a user-interface based on this stages of user activity model. These concepts are semantic distance of execution and evaluation, articulatory distance of execution and evaluation, and inter-referential input and output. They did not propose how such concepts could actually be measured. We were interested in determining whether existing USI evaluation techniques addressed USI problems in all stages of the HCI cycle and whether they got at issued of distance.

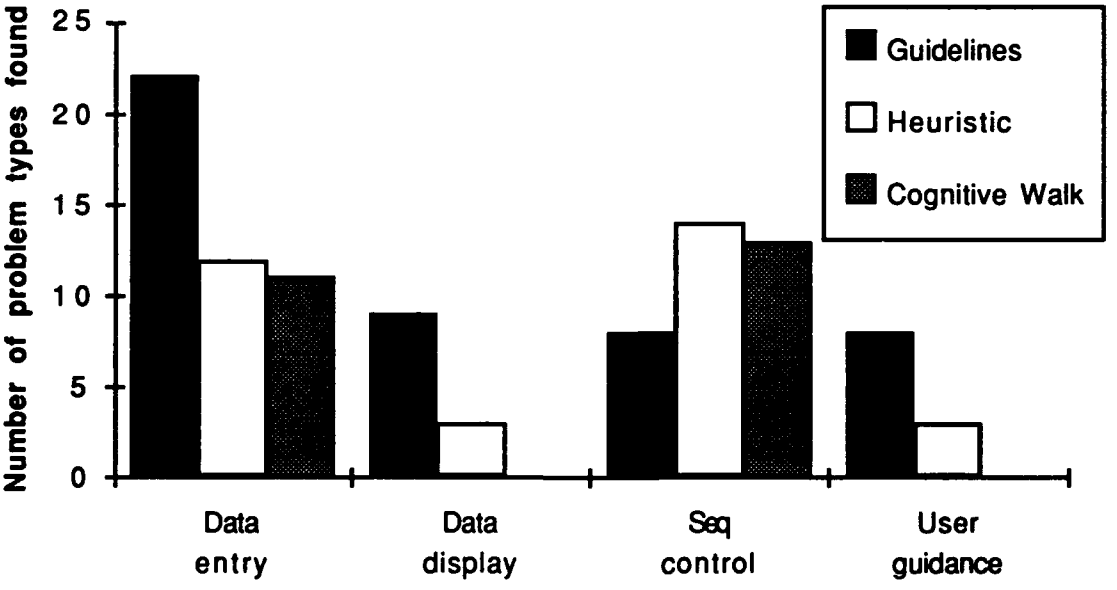
PROCEDURE

We had USI experts apply three evaluation techniques to a single prototyped scheduling system. The scheduling system has a direct manipulation, graphical user-interface style and was implemented on a Sun Workstation using Motif. Each evaluator received the same training on how to use the system. Typical tasks were demonstrated by a system designer. All problems predicted with each technique were recorded. The three techniques applied were an evaluation against the *Guidelines for Designing User Interface Software* (one evaluator), the heuristic evaluation (two independent evaluators whose results were later combined), and the cognitive walkthrough technique (one evaluator and one system designer working as a team). A USI guideline is a tested principle, ground rule, or rule of thumb for the design of the USI. Guidelines are necessarily general because they are applicable to many different kinds of systems. Heuristic evaluation involves having a USI expert or group of experts study an interface, and based on experience and training, identify potential areas of difficulty. The cognitive walkthrough technique attempts to extract design and evaluation guidance from a formal theory of human-computer interaction (Lewis et al., 1990). Questions are answered against a set of tasks to be performed with a system. The main part of the walkthrough involves evaluating the ease of learning the proposed design or system for each particular task.

RESULTS

We assessed the types of problems found against the stages of user activity model and the four functional areas outlined in Smith and Mosier (data entry, data display, sequence control, and user guidance). The graphs below show the number of problems each method found broken out by stage of user activity in the first graph and functional area in the second graph. There may be overlap between problem types found by methods.





DISCUSSION

We assessed the types of problems found by three structured judgement techniques against the Norman model of human-computer interaction and against the functional areas defined by Smith and Mosier. It was found that guidelines identified the most problem types overall, followed by heuristic evaluation, with cognitive walkthrough finding the least. Determining the number of problems found was not, however, the focus of this study. The point was to determine the range of problem types addressed by the different methods. We found that:

- guidelines and heuristic evaluation techniques addressed all of the stages of HCI at some level while the cognitive walkthrough addressed fewer stages;
- the cognitive walkthrough method found only one problem type for the whole evaluate cycle (last three stages);
- all of the techniques found the most problem types in the action specification stage;
- the guidelines and heuristic technique had the most overlap of any of the methods;
- of those problems found only by guidelines and heuristics, only in the action specification stage was the percentage of problems found uniquely by heuristics greater than those found by both; and
- overall all the methods were weak in measuring semantic distance on both the execution side (intention formation stage) and the evaluation side (evaluation stage).

We have tried with this study to carry the assessment of USI evaluation techniques one step beyond the most recent work in this area. This study indicates that current structured judgement evaluation methodologies are lacking when it comes to assessing the effects of the DM, graphical-style interface on all stages or functional areas of HCI. Current evaluation techniques and training received by USI evaluators are still deeply influenced by the large amounts of research in the text-based, data entry style displays. This provides a good evaluation for only one part of the interface. There is a lack of understanding and guidance on assessing the intention formulation stage and the entire evaluation side of the HCI activity cycle. To begin assessing the concept of semantic distance for intention formation, techniques would need to assess:

- whether users are allowed to work at the level they are thinking,
- the number of actions to accomplish a single goal, and
- whether the user is likely to have knowledge of the correct sequence of actions for a single goal.

For the evaluation cycle, every icon, display object, and action needs to be evaluated as to whether it has meaning to the user and is at the level the user thinks. The feedback to every user response needs to be assessed as to whether the user can now determine whether their goal was met at every level. The concept of level is important to HCI and is addressed somewhat by Norman (1986). Users have many levels of intentions, and subsequent levels of action specification. When performing an evaluation, all of these levels need to be understood and the required input and output assessed accordingly. For a single task, there could be a task level intention, a series of sub-task intentions, each with a sequence of actions to accomplish each sub-task intention, and an individual action level. Cognitive walkthrough seemed to work well only at the evaluation of the single action level, neglecting the higher levels.

Work also needs to continue on models of the HCI process. The inability to easily compare results across studies which look at the effectiveness of different evaluation techniques points to a need for a general framework within which evaluation methodologies can be compared. By using the framework suggested by the stages of user activity model to compare evaluation techniques, a more structured and cognitive-based approach to comparing evaluation techniques is possible. It too, however, could use some more detail, for example, in making the different levels more explicit.

In the third phase of this research program, we plan to investigate whether it is possible to obtain evaluations of semantic distance and better assessments of the other stages of user activity from usability studies. A key to this type of evaluation is understanding user's goals and previous knowledge which implies a great need for user participation. We have high hopes that proper analysis techniques applied to usability study data can provide us with assessments on the directness of the user interface design.

ACKNOWLEDGMENTS

We would like to thank the members of the USI Design and Evaluation group who participated in this study by performing the predictive evaluations: Tim Aiken, Janet Blackwell, and Linda Hoffberg. We would like to thank Nancy Goodwin for her numerous reviews of documentation and her helpful comments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1

INTRODUCTION

The focus of the Measures of User-System Interface Effectiveness project is to study and validate methodologies for measures and analyzing the overall effectiveness of user-system interfaces (USI) for task performance. There is an increased emphasis on user-centered system design which involves designing a system from a user's perspective, where the concepts, objects, and actions embodied in a system closely match the user's task concepts, objects, and actions. This paper documents the results of a study comparing three USI evaluation techniques to assess whether they provide adequate evaluation of the newer graphical, direct manipulation style interfaces. The types of problems identified by each method were classified according to two generic models of human-computer interaction (HCI). This part of the research was just one phase of an overall research program.

1.1 RESEARCH PROGRAM

The plan for the entire FY92 MSR project is shown in figure 1. The first step was to review models of HCI, review existing USI evaluation techniques and data analysis tools, and to derive HCI-effectiveness measures based on the models of human cognition and HCI. Volume I of this MSR report series documented the results of this first phase of the research (MTR 92B0000047). We identified the need for a review of existing USI methodologies in light of the newer graphical, direct-manipulation style interfaces and the need to develop measures which reflect how well these interfaces support the users and represent their task domain. This led to the second phase of the research where predictive evaluation techniques were both assessed and used to predict where users might encounter cognitive difficulties during task performance. These evaluation techniques were applied to the Military Airspace Management System (MAMS), a prototype of a military airspace scheduling system. This system served as our application system for the entire study and was selected because it has a graphical, direct manipulation style interface. This phase of the research is documented in this report.

The third and final phase of the research program will involve reporting on the results of a user-based evaluation which was just recently conducted. Data was collected on schedulers using the prototyped system. From the data, we will try to extract measures which reflect the predicted cognitive difficulty. We will also try to identify problems we predicted from phase 2 but could not readily identify via collected data, as well as problems which show up in the data but we did not predict. In this way, we will identify the usefulness of predictive techniques and identify to what degree cognitive aspects of HCI can be assessed from user-based evaluations. We will also attempt to validate the cognitive and HCI models based on the collected data.

This volume of the MSR MTR series documents the results of applying three evaluation techniques to predict user performance. We were particularly interested in assessing which of Norman's stages of user activity were evaluated by each technique and if any got at the concepts of semantic and articulatory distance. Below, we briefly review Norman's and Hutchins, Hollan and Norman's theories on stages of user activity and the concepts of semantic and articulatory distance (they are explained in greater detail in MTR 92B0000047). Then we discuss the three evaluation techniques used, the study, and the results.

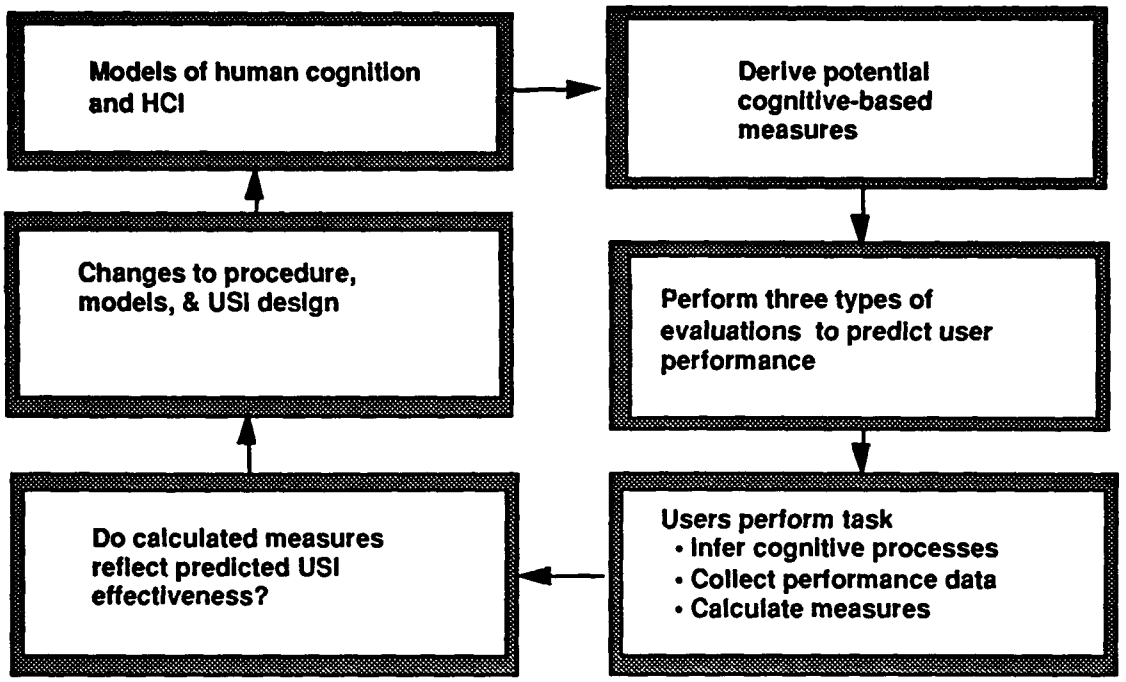


Figure 1. Overall Research Plan

1.2 FORMAL FRAMEWORK OF HUMAN-COMPUTER INTERACTION (HCI)

Norman (1986) has proposed a descriptive model of human-computer interaction (figure 2) which addresses some of the issues which contribute to a feeling of directness in a graphical, direct manipulation (DM style) interface. His model describes a user's interaction with the computer as occurring in seven stages: establishing the goal, forming the intention, specifying the action sequence, executing the action, perceiving the system state, interpreting the system state, and evaluation of the system state with respect to the goals and intentions. The first four stages encompass the execution cycle while the last three stages encompass the evaluation cycle.

Forming an intention is the activity that specifies the meaning of the input expression that is to satisfy the user's goal. The action specification prescribes the form of an input expression having the desired meaning. These two activities are psychological activities. The form of the input expression is then executed by the user on the computer interface and the form of the output expression appears on the display, to be perceived by the user. Interpretation determines the meaning of the output expression from the form of the output expression. Evaluation assesses the relationship between the meaning of the output expression and the user's goals (Hutchins, Hollan, and Norman, 1986). The last two stages are also psychological activities.

Based on this model, Hutchins et al.(1986) proposed concepts related to the directness of engagements for a user interface. These are semantic distance of execution and evaluation (the intention formation and evaluation stages), and articulatory distance of execution and evaluation (the action specification and interpretation stages). Semantic directness involves matching the level of description required by the interface language to the level at which the person thinks of the task. Is it possible to say what one wants with this language? Can the things be said concisely? Can one easily evaluate whether their intention was met? Articulatory directness involves how well the form of an expression relates to the meaning on both the input and output side (Hutchins et al., 1986).

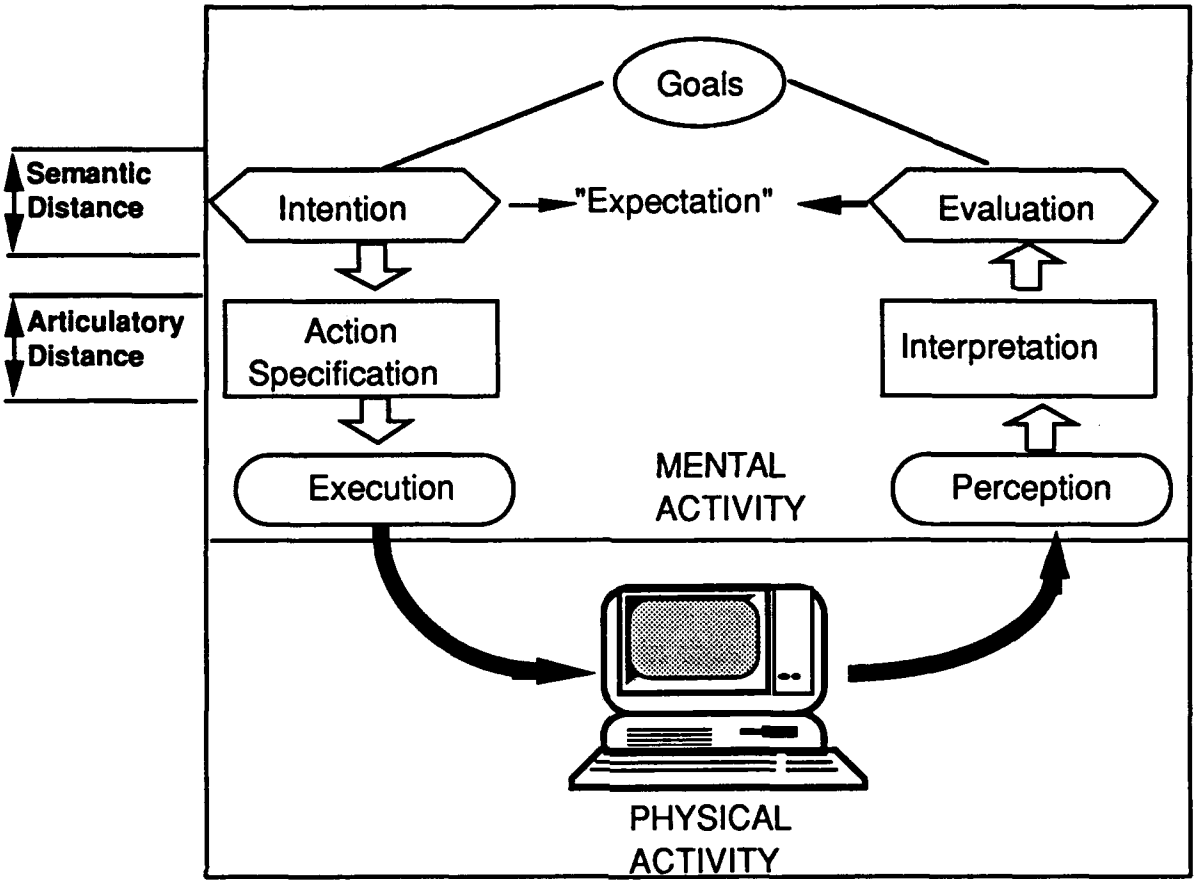


Figure 2. Semantic and Articulatory Distance (Hutchins et al., 1986)

1.3 SUMMARY OF EVALUATION METHODS

Our previous evaluation of USI evaluation methods resulted in the following taxonomy:

Evaluation Category	Requirements	Evaluation Technique
User-based Evaluations	Requires a system or prototype, users, and a researcher.	Usability study Experiments HCI research
Analytic Evaluations	Requires a USI design, and an expert on the analytic technique.	Keystroke Level Model GOMS Production Systems Grammars
Structured Judgement Techniques	Requires a system or prototype, and USI expert.	Guidelines Heuristic Evaluation Cognitive Walkthrough

Based on the literature reporting the effectiveness of the different techniques (e.g., Olson et al.,1990), and our own personnel experiences, we eliminated analytic evaluations from our review at this time. We are addressing user-based evaluations in phase 3 of the research program. For this phase, we concentrated our efforts on structured judgement techniques and selected guidelines, heuristic evalutions, and cognitive walkthrough. Structured judgement techniques are

useful and effective when applied during the early stages of design. We were very familiar with the guidelines and heuristic evalution methods but were less familiar, with the newer cognitive walkthrough technique.

1.3.1 USI Guidelines

A USI guideline is a tested principle, ground rule, or rule of thumb for the design of the USI. Guidelines are necessarily general because they are applicable to many different kinds of systems. There exist many documents which contain general guidelines to aid in the development of a good USI. One of the more complete set of guidelines is the "Guidelines for Designing User Interface Software" by Smith and Mosier (1986). Smith and Mosier contains 944 guidelines divided into 6 functional areas: data entry, data display, sequence control, user guidance, data transmission, and data protection. An example of a guideline is:

"Format a menu to indicate logically related groups of options, rather than as an undifferentiated string of alternatives."

Applying general USI guidelines can be difficult as they offer the application developer little guidance concerning where, when, and how to use them. Performing an evaluation against these guidelines can also be difficult. Guidelines need to be assessed as relevant to a particular system and a judgement made on the system's compliance with each applicable guideline.

1.3.2 Heuristic Evaluation

Heuristic evaluation, according to Jeffries et al. (1991), involves having a USI expert or group of experts study an interface, and based on experience and training, identify potential areas of difficulty. Typically, a heuristic evaluation involves a USI expert, who has knowledge of good user interface design principles internalized, reviewing or looking at an interface and identifying potential areas of difficulty. There was no general agreement on the definition of a heuristic evaluation in the literature although many people believe it is the most commonly used technique in practice. Nielsen and Molich (1990) defined heuristic evaluations as evaluators looking at the interface and passing judgement according to their own opinion; the evaluators are not necessarily USI experts. Both Jeffries et al. and Nielson et al. found that heuristic evaluations are more effective when a group of independent evaluators is used, as compared to a single individual. The heuristic method is commonly used at MITRE and is probably the type of evaluation we are most frequently asked to perform.

1.3.3 Cognitive Walkthrough

The cognitive walkthrough technique attempts to extract design and evaluation guidance from a formal theory of human-computer interaction (Lewis et al., 1990). The method is based on a theory of exploratory learning and is a list of theoretically motivated questions about the USI. The questions are answered against a set of tasks to be performed with a system. The main part of the walkthrough involves evaluating the ease of learning the proposed design or system for each particular task (Lewis et al., 1990). It was primarily intended for walk-up and use interfaces (e.g., automated teller machines). The cognitive walkthrough evaluation form for a single action is shown on the next page. The answers to the questions are based on a text-editing task of spell checking a document using the Macintosh.

The cognitive walkthrough technique works back from the designer's design toward the user's likely goals. The first step in the technique is to describe the level of knowledge of the user population. You might, for instance, assume they have familiarity with the Macintosh. Next, you are asked to list the goals a user is likely to have for completing a particular task. These are probably

Walkthrough Start-up Form _____

I. Task description

Check the spelling of file "my.paper"

II. Initial goals (Goal structure a user is assumed to have)

1.0 Start the word processor
2.0 Load the "my.paper" file
3.0 Run the spelling checker

Next Action #: _1_ **Description:** _Double click on word processor icon_

I. Correct goals

1.0 Start the word processor
    1.1 Double click on word processor icon
2.0 Load the "my.paper" file
3.0 Run the spelling checker

II. Problems forming correct goals

A. Failure to add goals. __30__ %

B. Failure to drop goals. __0___ %

C. Addition of spurious goals. ___0__ %

No-progress impasse. _____ %

D. Premature loss of goals. ___0__ %

Supergoal kill-off. _____ %

III. Problems identifying the action

A. Correct action doesn't match goal. ___90__ %
The action of double clicking on an application icon is not intuitive.

B. Incorrect actions match goals. ___60_ %
Users may select "Open" from the desktop File Menu.

IV. Problems performing the action

A. Physical difficulties. __30__%
Some individuals have difficulty double clicking.

B. Time-outs. _____%

high level goals. Then, working off of a list of the correct actions needed to perform that goal with the given system, you write down the goals which the user would have to have generated to think to perform that action. This is compared to the users set of initial goals. Then you assess whether users are likely to have had the goal, or are likely to have deleted a goal which they may have initially had but was not required, etc. Assessments are made by indicating the percentage of users you think might experience a problem and an explanation of the potential problem.

The next part involves assessing the actual actions for completing the goal with the system. The action is compared to the goal and an assessment is made of whether there is an action-goal match. For instance, if the goal is to open a document and there is a menu command called "Open document", the action-goal match probability will be high. For the same example, if the action to open a document is to double click on the document, that is not as obvious and may be rated as causing some percentage of users difficulty. Here is where the user population description becomes relevant. If you can assume the users are Macintosh-literate then this may be an obvious action. Next, you examine the interface for any false-action matches. This means are there any other actions the user could take at this point in time which might appear to meet their goal. For instance, if the goal for a Macintosh user is to change the selected printer and under the apple menu there is both a chooser function and a control panel function, you might predict that some users would think the control panel function would be correct. Finally, there are some questions on the physical difficulty of performing certain actions, such as having a time-out period. In addition to the paper forms, there is an on-line, HyperCard version of the cognitive walkthrough evaluation (Rieman, et al., 1990).

1.4 STUDIES COMPARING EVALUATION METHODOLOGIES

The number of available evaluation techniques raises questions on which techniques are "best", or the more logical question "which types of problems can each technique identify and when can each be used?" Recently some studies were performed comparing the ability of various techniques to identify user-interface problems. In Jeffries et al. (1991), four USI evaluation techniques were applied to a single software product: heuristic evaluation, software guidelines, cognitive walkthroughs, and usability testing. The authors felt that little was known about how each of these techniques work and what kinds of interface problems they are best-suited to detect.

Each technique was applied by a separate group of people at Hewlett Packard labs under realistic conditions. The package evaluated was HP-VUE. Each group used a common form to record USI problems. Four USI experts performed the heuristic evaluations. The usability test consisted of six subjects performing a set of 10 user tasks. The guideline group used a set of 62 internal guidelines applied to the portions of the system used to complete the 10 tasks. The cognitive walkthrough was performed by a group of evaluators on the same 10 user tasks.

Results showed that heuristic evaluations by a group of experts identified the most problems at low cost but required the input of several USI experts, who may not always be available. Many nonsevere problems were also discovered via this technique. Usability testing was the next most successful technique, also uncovering problems ranked as having the highest severity, but at a higher cost. The use of guidelines and cognitive walkthroughs each had advantages and disadvantages but were not as useful for evaluating this particular application; guidelines found recurring but not necessarily serious problems. The heuristic and walkthrough group also seemed to use more subjective criteria in their evaluations. Although heuristic evaluation did very well, it was noted that several skilled people were required to do the evaluation. Problems occurring as a result of user error were found only with the usability study.

Lewis et al. (1990) compared the cognitive walkthrough technique to results obtained by empirical (user-based) testing. Four evaluators each performed independent cognitive walkthroughs for two tasks for four interfaces to a mail messaging system. Twenty unique problems had been identified across all evaluators. The authors claim that with this technique, a group of evaluators can detect almost 50 percent of the problems that would be revealed by a full-scale (user-based) evaluation. They also feel that the walkthrough methodology requires a limited investment in resources.

None of the comparisons described above attempted to determine what types of problems each techniques found. So, we next applied the three structured judgement techniques to a graphical, DM-style interface to identify the types of problems each technique was capable of finding along with the areas of the HCI process addressed. Problem areas were classified by the stages from the user activity model as well as by the functional areas outlined in Smith and Mosier (1986). We also discuss whether the existing USI evaluation techniques address the newer concepts of interface directness such as those proposed by Hutchins et al.(1986).

1.5 SUMMARY

Evaluation techniques were reviewed and three were selected to be applied to a graphical, direct-manipulation style interface. Models of HCI were reviewed and the stages of activity model and its related concepts was selected as being a good candidate against which to assess the types of problems identified by evaluation techniques. The purpose was to determine if existing evaluation techniques addressed concepts important to the directness of engagements experienced by users of graphical interfaces.

SECTION 2

METHOD

Three structured judgement techniques were applied to a prototype airspace scheduling system.

2.1 PROCEDURE

Five human factors professionals with USI and evaluation experience participated in the comparison of structured judgement techniques. Every evaluator received a standardized briefing on the prototype system, were walked through preselected typical tasks, and asked to identify USI problems. One evaluator was assigned to evaluate the interface against the Smith and Mosier guidelines; he was very familiar with the guidelines and this method. A checklist was made by the evaluator from the guidelines using the four sections out of the six which were applicable to the test system. The evaluator then looked for instances where the USI violated any of the guidelines and noted that the USI was not compliant with a particular guideline. Often violating one guideline would mean non-compliance with other related guidelines. Two evaluators were assigned the heuristic evaluation method; each performed an independent evaluation. The heuristic evaluators recorded problems using any method they chose. Two evaluators were assigned to work together using the cognitive walkthrough method. One was a member of the prototype design team and one an independent evaluator. The cognitive walkthrough evaluators used the Automated Cognitive Walkthrough (CW) HyperCard stack (Rieman et al., 1990). This is an on-line checklist which leads the user through the CW form and the problems noted are typed into the system. Upon completion, the program prints out a summary of all the problems identified for each task.

2.2 MAMS

The Military Airspace Management System (MAMS) is being developed as a tool for effective scheduling and a means of collecting and reporting airspace utilization data. Using the MAMS system, DOD airspace managers can quickly request and approve missions in both local and remote special use airspaces by means of graphical user interface.

The MAMS prototype is being developed as a vehicle to help define the requirements of an airspace management system, validate the system architecture, and refine a graphical user interface to the system. The prototype development was scheduled to proceed for eighteen months. The initial focus was to address the user interface and unique scheduling problems associated with airspace management. There are over 200 military airspace scheduling organizations, each with unique requirements and site-specific procedures for scheduling and managing their airspaces. Consequently, a wide variety of scheduling methods and computing tools presently exist. User participation was imperative to define baseline scheduling methods to meet user needs.

To ensure that the development of the prototype was rapid, the initial prototype was built using an existing scheduling system, the MITRE-developed Range Scheduling Aid (RSA). The Range Scheduling Aid's graphical user interface has the look of a Gantt chart, and allows use of a mouse to directly manipulate the time periods represented by color-coded screen icons. These basic concepts were carried over to the MAMS prototype.

The main screen for the MAMS prototype (see figure 3) presents a menu bar, the screen start date, and a time scale at the top of the screen. The screen is divided into a number of horizontal areas called "panes," each of which is associated with specific airspaces selected by the user.

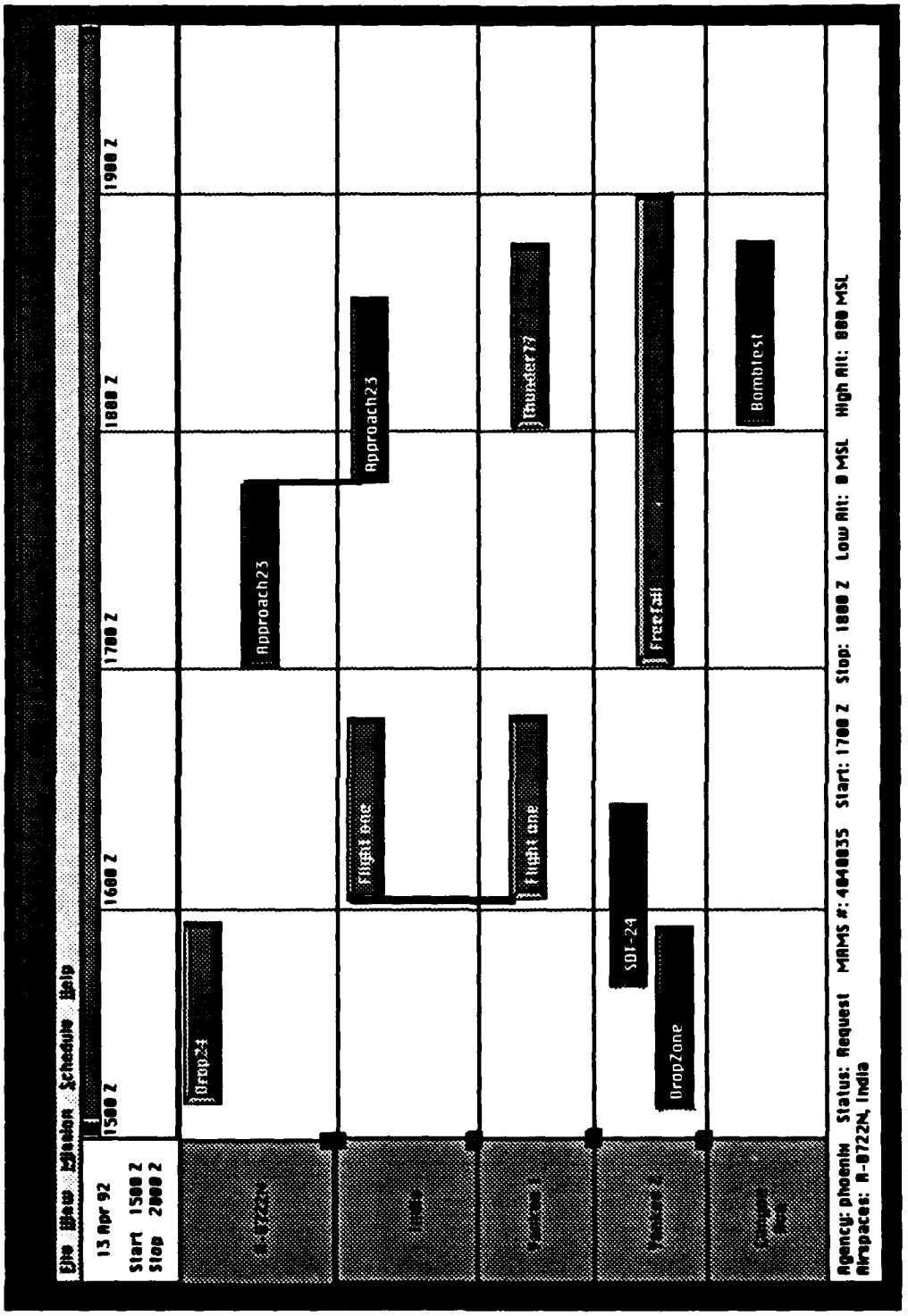


Figure 3. MAMS Main Screen

Inside each pane are mission requests or scheduled missions, which are represented by colored, bar-shaped icons with a fixed height and a length proportional to the length of the mission. A mission identifier or name is displayed within the mission icon. To change the time of a mission request or scheduled mission, the user simply drags the icon with the mouse. The pane at the bottom of the MAMS screen is used to display a description of the currently selected mission to the user.

Pop-up dialogs are used to obtain input from the user. The system opens a specific pop-up dialog when information is required. The pop-up dialog is an electronic version of a paper form and often appears as the result of a user's menu choice selection. Text can be typed directly into portions of the form which are colored white, while other data may be entered through the use of radio buttons or by making selections from option menus as described earlier. The MAMS system contains a number of pop-up dialogs. The major dialogs which underwent testing are described below.

The Folder dialog (figure 4) allows the creation and editing of airspace groupings, referred to as folders. These folders can then be used in the General Layout menu to set up the MAMS display with a preset group of airspaces.

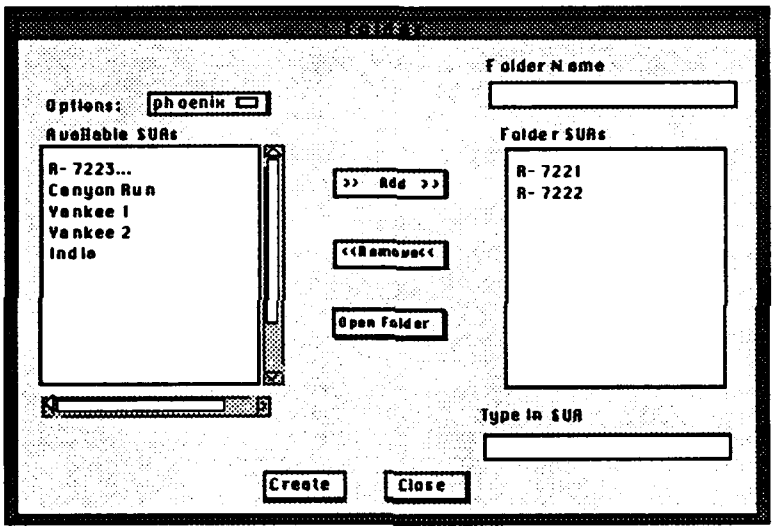


Figure 4. Folder Dialog

The Change Layout dialog (figure 5) is used to select the SUAs to be displayed on the screen. When Change Layout is selected, a dialog box is displayed which allows the user to scroll through the list of SUAs available for screen viewing. The SUA list is grouped by scheduling agency and includes any folders defined by the scheduling agency. A direct entry feature lets the user type the name of a selected SUA to be displayed on the screen. The system then searches for the airspace whose name most closely matches the user entry. If a match is found, the airspace will be added to the main display. Otherwise, the system will not respond.

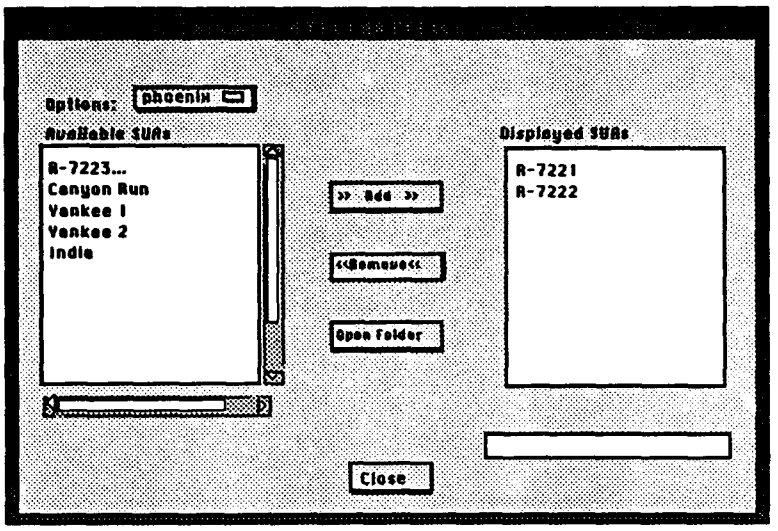


Figure 5. Change Layout Dialog

Selection of the Set Date and Times dialog (figure 6) allows the start date, start time and screen display duration to be manipulated. MAMS supports the entry and display of times in Zulu or local formats. By entering the time and the time format in the Time field of the Set Date and Times dialog, the user is able to display times in either format.

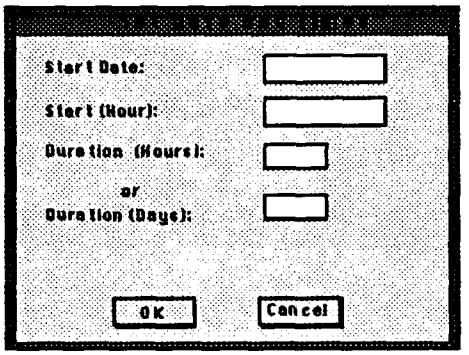


Figure 6. Set Date and Times Dialog

The Create New Mission dialog (figure 7) is used to reserve airspace. The dialog has a button to create either a mission request or an approved mission. Only users who have authority over the airspaces specified in the SUA data entry field can create an approved mission.

The Edit Mission dialog (figure 8) allows the user to edit a mission either located in an airspace over which the user has scheduling authority or for which the user was the original requester. Otherwise, the mission data may only be viewed using this option. If the mission has not been approved, the requester is allowed to edit all of the mission data. Once the mission is scheduled however, the requester is only allowed to edit the data not associated with the SUA information. The scheduler, on the other hand, is only allowed to edit the SUA information, whether or not the mission has been approved. Editing a mission is accomplished by first selecting the icon that represents the mission request or approved mission with the mouse and then selecting the Edit/View Mission option from the Mission menu.

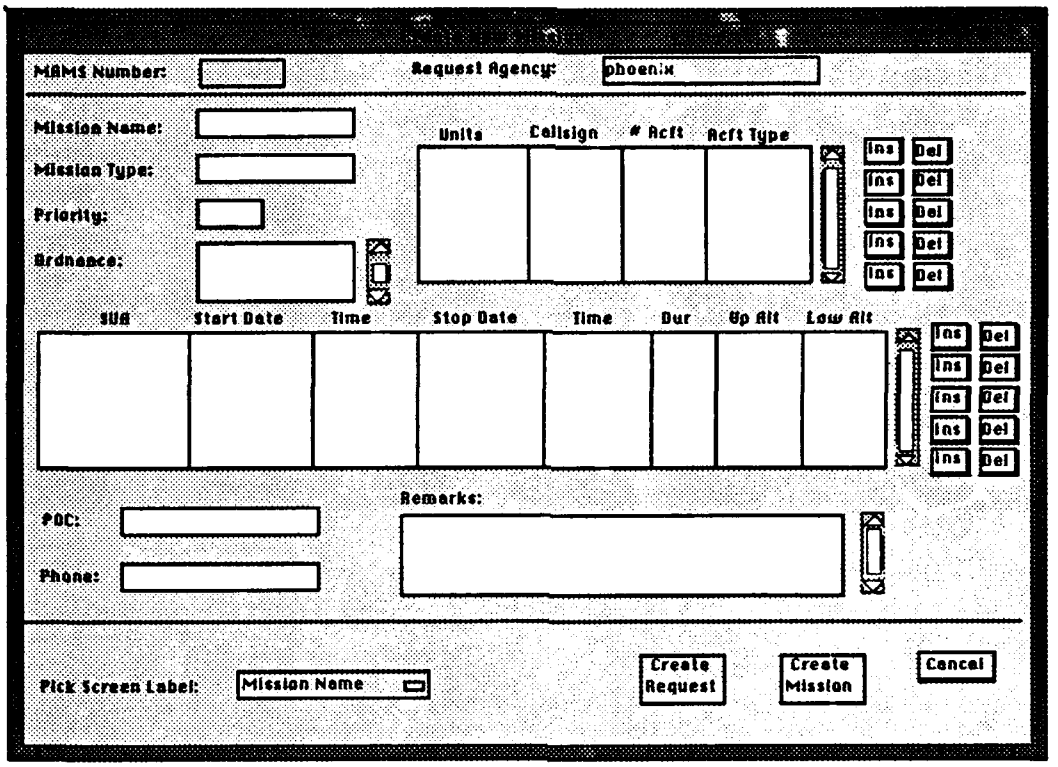


Figure 7. Create New Mission Dialog

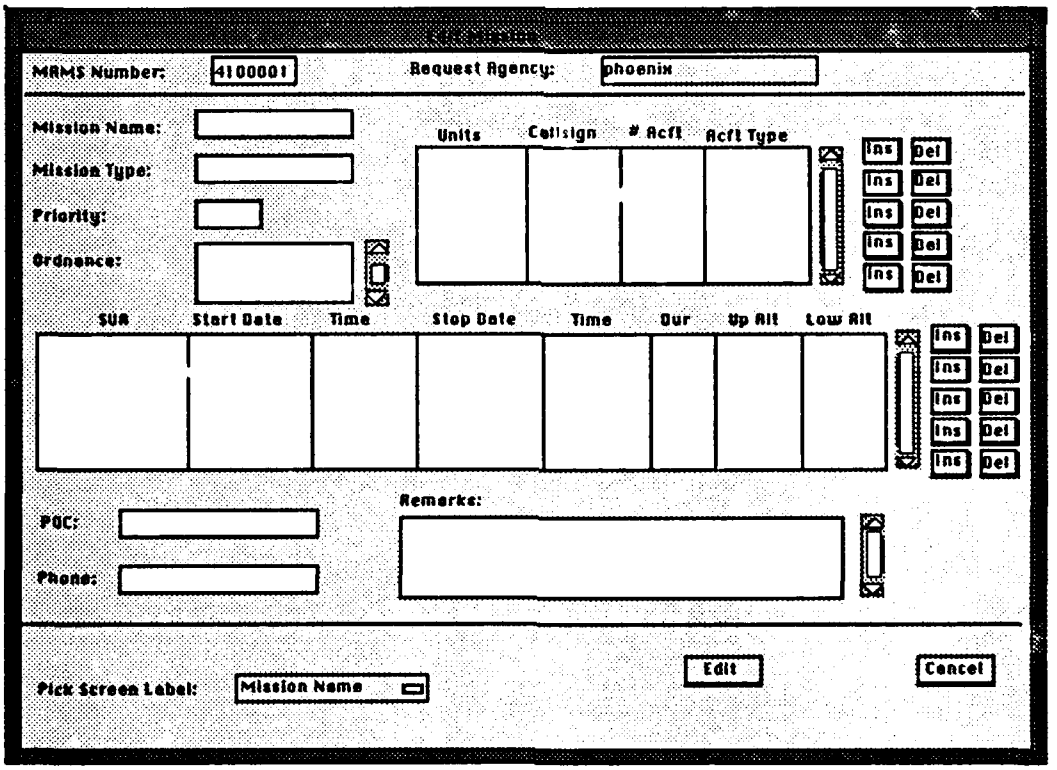


Figure 8. Edit Mission Dialog

The Find Mission dialog (Figure 9) allows the user to find particular mission request(s) and/or approved mission(s) by entering specified criteria that describes the mission. The prototype will present a list of the missions which meet the user specified criteria. The user then has the option to adjust the main display set-up times and SUAs displayed to view the selected missions by pressing the Change Screen button, to Copy the mission, or to View the mission data.

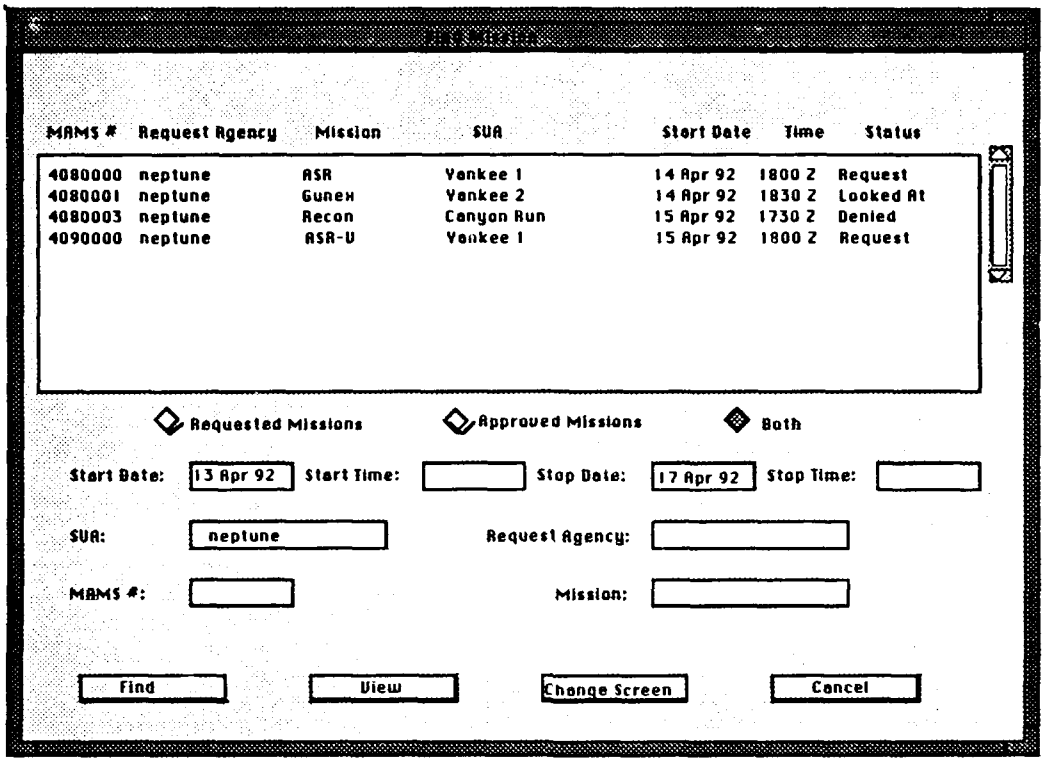


Figure 9. Find Mission Dialog

The Pending Request List (figure 10) presents a list of pending requests that spans the time entered in the Start Date field to the time entered in the Stop Date field. The list can be altered by the user to include only requests in a specific airspace and/or made by a specific requesting agency.

The Reports dialog (figure 11) allows users to view four types of reports on the screen and to send them to the printer. The reports include information related to a selected mission, missions in an SUA, missions requested by a given agency, and a utilization report for a specified SUA.

2.3 TASKS

The following tasks were covered in training and demonstrated to the evaluators: Create, Approve, Deny, Edit a mission/request, create a folder, edit a folder, and print a report. The task description followed for the cognitive walkthrough method is provided in Appendix B.

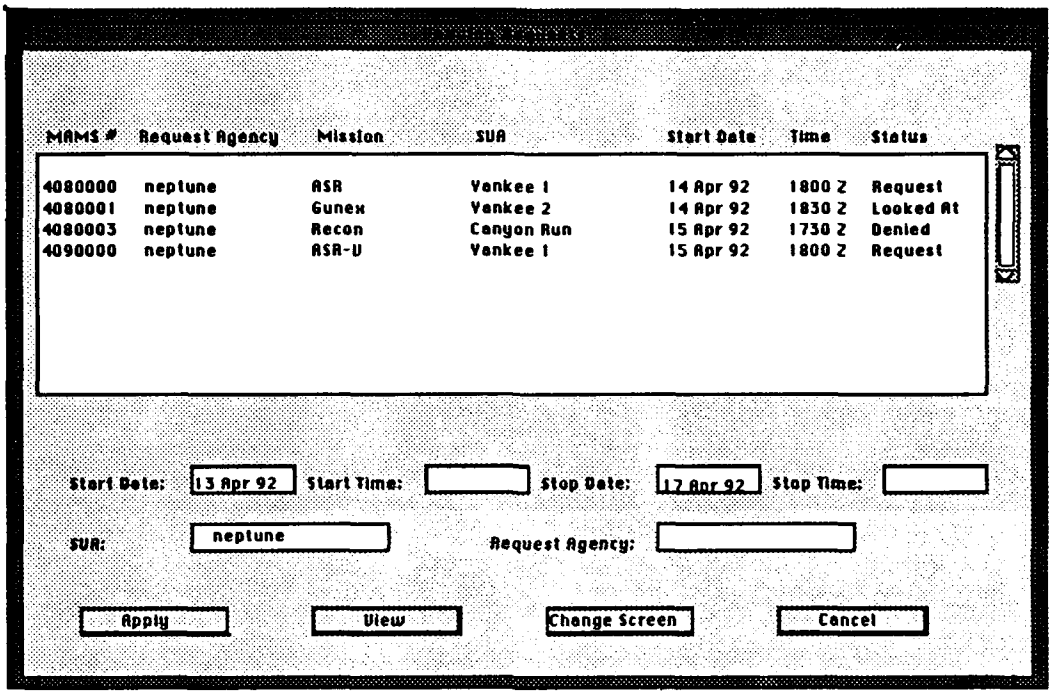


Figure 10. Pending Request List Dialog

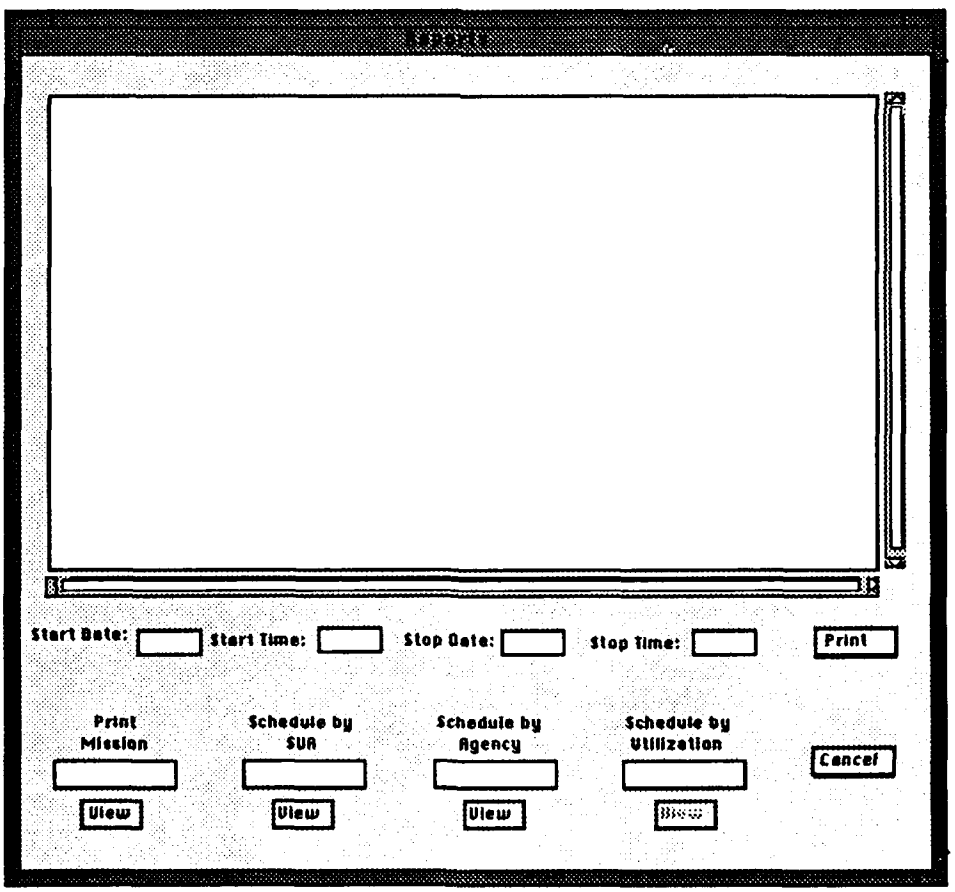


Figure 11. Reports Dialog

SECTION 3

RESULTS

Following are the results from the application of the three evaluation techniques to the MAMS prototype.

3.1 TIME FOR EACH EVALUATION

Evaluation Method	Time for Evaluation
Heuristic 1	2 hours 10 minutes
Heuristic 2	2 hours 45 minutes
Guidelines	9 hours 30 minutes
Cognitive Walkthrough	8 hours 30 minutes

3.2 PROBLEM FILTERING

Table 1 summarizes the numbers of problems identified after various levels of filtering. The second column shows the number of raw problem reports generated by each of the evaluators. The third column shows the number of problem reports after being filtered. Problem reports were eliminated for various reasons: evaluator error/confusion about the system, problem reported a known system bug which was not a USI design problem, problem was not stated in the form of a problem but rather as an alternative design solution, or the problem related to pieces of the system which were not yet implemented in the prototype (e.g., guidelines applying to the design of the help system were eliminated because the help system implementation was not part of the prototype). Finally, the problems were filtered for redundancies within evaluation methods, instances of the same problem type were grouped into one problem type category, and the results from the two heuristic evaluators were grouped together. An example of grouping instances into a problem type is several instances of not disabling non-active menu items or buttons were reported. These were lumped into the problem type non-active options not disabled. Specific instances were still recorded but were not counted as different problem types.

Table 1. Initial Number of Problems Identified and Subsequent Filtering

Evaluation Method	*Number of Raw Problems*	*Number of Problems*	*Number of Problem Types*
Heuristic 1	47	29	16
Heuristic 2	32	28	26
Combined Heuristic results	--	--	32
Guidelines	*216	*113	**47
Cognitive Walkthrough	46	43	24

*applicable guidelines the system was not in compliance with
**many guidelines could be applicable to a single problem type

3.3 RESULTS BY STAGE OF USER ACTIVITY

Each problem type found was then allocated to a stage of user activity. Mapping the problem types to stages was easier in some cases than others, given the rather vague definitions of semantic and articulatory distances. The specific problem types and their resulting classifications are provided in Appendix A. All problem types concerned with issues in some way related to how easily or whether the user would be able to express an intention were classified as intention formation. These included problems like lack of an "undo", inability to apply an action to multiple objects at once, lack of indication of mandatory fields which could imply more information is required than really is needed, the need to remove default data before being allowed to fill in actual data, and lack of shortcuts for specific actions.

All problem types concerned with issues in some way related to the form of the input expression were classified as action specification. These included problems on labels (poorly worded, inconsistent, or misleading), prompts, cues, indications of editable fields, abbreviations, indications of acceptable data formats, making fields active, specific instances which would cause wrong action to be performed or selected, areas where sequences of actions weren't obvious, allowing non-current actions to appear active, inconsistency in actions, lack of punctuation, location of menu items not obvious, etc. All these types of issues were thought to be related to how well the user's intention mapped to the required action.

All problem types concerned with issues in some way related to the execution of the input expression were classified as executes. These included problems on allowing users to change or remove system default values and keeping these values, overly long and unformatted numbers, lack of input focus when windows appear, lack of automatic justification of data, difficulty with selecting missions when timeline is large, cursor not positioned usefully or consistently, mandatory fields not put first, difficulty with finetuning mission icon position, lack of consistent location for buttons, scheduling scroll bar arrows too small, cursor not placed at most frequently used option in a list, difficulty in selecting from hierarchical menus, lack of notification when keyboard is locked.

All problem types concerned with issues in some way related to the perception of the computer output were classified as perception. These included problems like inconsistent data labels, long numbers not formatted, lack of cues for row scanning, nonuse of mixed-case fonts, extremely small mission icon labels and grid line overlaps labels when timeline is large, difficulty in seeing tapes in simultaneous missions, poor visual feedback, inconsistent display formats and design standards, lack of blink coding for urgent items, and cursor not readily distinguishable from background items.

All problem types concerned with issues in some way related to the interpretation of the computer output were classified as interpretation. These included problems like poor grouping of data entry fields and data items, lack of or poorly placed data unit labels, lack of names and titles on certain items, lack of a standard symbol for prompts, overstrike vs. insert mode not distinguishable, blue color too saturated since not critical data, poor visual feedback, inconsistent display formats and design standards, lack of blink coding for urgent items, default system selections not indicated to user, error handling, no dictionary provided of abbreviations and codes, and error messages incorrectly worded.

All problem types concerned with issues in some way related to the evaluation of the computer output were classified as evaluation. These included problems like no error messages when enter invalid data, no feedback for successful actions, and lack of feedback especially when system is working slowly.

The classification process, as with the definition of problem types, contained some ambiguity and some classifications could be debated. We are confident however that the resulting classifications were reasonable and suitable for our purposes.

The resulting number of problem types for the classification scheme are shown in table 2. In table 2, the total column shows how many problem types were found for each stage by each method. Problems that were found only by a single method are shown in the next column labeled unique; the instances where a single problem was identified by two or three methods are also shown. Some problem types mapped to more than one stage and three problem types did not map to any stage; these were not counted.

Table 2. Number of Problem Types Found for Each User Activity Stage by Method

Evaluation Method	*Intention*		*ActionSpec*		*Execute*		*Perception*		*Interpret*		*Evaluate*	
Number found by:	*Total*	*Unique*	*Total*	*Unique*	*Total*	*Unique*	*Total*	*Unique*	*Total*	*Unique*	*Total*	*Unique*
Guidelines	3	0	14	6	11	10	10	8	10	6	3	0
Heuristic	3	1	18	9	2	1	3	1	6	2	3	0
Cognitive Walkthrough	3	2	18	16	2	2	0	0	1	1	0	0
Guide & Heur		2		7		1		2		4		3
Heur & CW		0		1		0		0		0		0
Guide & CW		1		0		0		0		0		0
Guide/Heur/CW		0		1		0		0		0		0
Total number of problem types found		6		40		14		11		13		3

The majority of the problems identified, forty, were classified as action specification problems. Action specification is the activity that prescribes the form of an input expression having the desired meaning. Problems in the stages of execute, interpret, and perception were the next most frequent with 14, 13 and 11 problem types found. Finally, intention formation and evaluation problems were the least frequently found with 6 and 3 problem types, respectively.

Of the three methods, it appears that guidelines were more likely to find problems for each of the six stages, with heuristic next. Cognitive walkthrough found a total of only 1 problem type for the last three stages. CW did very well in the action specification stage, however, tying with heuristics for the most problem types found in this stage, and having the greatest number of unique problems found. For the intention and evaluate stages, there was a large amount of overlap of problem types between the guideline and heuristic methods. The problem types found by the cognitive walkthrough rarely overlapped with problem types found by other methods.

We next attempted to further classify the problem types by whether they applied to objects or operations but found this to be too difficult. For example, two problems identified via the guidelines and heuristic methods were "lack of indicators of acceptable data formats", and "entered data should be case insensitive". These were classified as action specification problems as they affect the ease of getting a form match with an input action but it is not clear if they would be considered object or operation mismatches.

3.4 RESULTS BY GUIDELINE FUNCTIONAL AREA

A final classification was performed by breaking out problem types by functional area as defined in Smith and Mosier (1986). This is shown in table 3; note that some problem types mapped to multiple categories. Results indicate that the most problem types found were in the area of data entry (34), closely followed by sequence control (30). Lagging far behind, making up about a fifth of the problem types identified, were problems in the areas of data display (10) and user guidance (8). These results are consistent with the stages classification results -- the data entry and sequence control areas tend to correspond with the action specification stage although it is not a one-to-one mapping. For example, some guidelines on data entry could be related to the perception or interpretation stages.

For individual methods, CW again found no problem in two of the four functional areas. For data entry, guidelines found the most problem types while for sequence control, CW and heuristic methods found the most problems, with CW finding the most unique problem types. For data display, guidelines far outdistanced the other methods; CW found no problems of this type and heuristics found only one unique problem. For user guidance, the heuristic method did not find any problem types not also found by guidelines and CW found no problems of this type. Heuristic and guideline methods again had the most overlap between methods.

Table 3. Number of Problem Types Found for Each Functional Area by Method

Evaluation Method	*Data Entry*		*Data Display*		*Sequence Control*		*User Guidance*	
Number found by:	*Total*	*Unique*	*Total*	*Unique*	*Total*	*Unique*	*Total*	*Unique*
Guidelines	22	12	9	7	8	4	8	5
Heuristic	12	3	3	1	14	9	3	0
Cognitive Walkthrough	11	9	0	0	13	12	0	0
Guide & Heur		8		2		4		3
Heur & CW		0		0		1		
Guide & CW		1		0		0		
Guide/Heur/CW		1		0		0		
Total number of problem types found		34		10		30		8

SECTION 4

DISCUSSION

We assessed the types of problems found by three structured judgement techniques against the Norman model of human-computer interaction and against the functional areas defined by Smith and Mosier. It was found that guidelines identified the most problem types overall, followed by heuristic evaluation, with cognitive walkthrough finding the least. Determining the number of problems found was not, however, the focus of this study. The point was to determine the range of problem types addressed by the different methods. We showed that guidelines and heuristic evaluation techniques addressed all of the stages of HCI at some level while the cognitive walkthrough addressed fewer stages. The cognitive walkthrough method found only one problem type for the whole evaluate cycle (last three stages). All of the techniques found the most problem types in the action specification stage. The guidelines and heuristic technique had the most overlap of any of the methods. Of those problems found only by guidelines and heuristics, only in the action specification stage was the percentage of problems found uniquely by heuristics greater than those found by both. Overall, however, all the methods were weak in measuring semantic distance on both the execution side (intention formation stage) and the evaluation side (evaluation stage).

4.1 GUIDELINES

Guideline evaluations are useful in that they force the evaluator to address all areas for which guidelines exist. This has an associated time expense but is very thorough *in the areas for which guidelines exist.* The problem is that there are not a lot of guidelines concerned with graphical, direct manipulation style interfaces, and if there were, they would necessarily be general. Guidelines, when applied directly, also do not necessarily consider task-based, goal-oriented user behavior. Thus they provide inadequate evaluation of semantic distance. Different types of techniques are needed to assess the new interface styles. It is interesting to note, however, that most interfaces are a compilation of interface styles. The prototype scheduling system had, for instance, many form-fill dialogue boxes. Guidelines did very well in evaluating this part of the interface.

4.2 COGNITIVE WALKTHROUGH

We had hoped that cognitive walkthroughs would have provided thorough evaluations of both semantic and articulatory distance of the execution side of the cycle. The questions on the failure to add/drop goals, additions of spurious goals, and premature loss of goals seemed like they would relate to whether the steps required by the computer to accomplish a goal matched the sequence of steps a user would expect to have to take. For instance, if the computer required many indirect actions for a single goal, it would be predicted that the user would fail to add these steps as interim goals. If the computer automatically performed a sequence of steps the user expected to perform manually and separately, the user would fail to drop goals. These questions seemed to address the semantically-related questions of "can I say it easily" and "does it do what I want it to do?" Yet only 3 problems for intention formation were found with this technique. It turned out that some of the add/drop/spurious goals occured at a low, action level and were classified as action specification issues. The CW technique does not make a clear distinction between actions and goals which makes its questions difficult to apply. Also, the technique does not ask questions on the overall complexity of actions to complete a single goal, rather the single goal is broken down into low-level steps and each of these are evaluated. It is also possible that the evaluators could not accurately know what the goals of the users would actually be and what knowledge they would have. Finally, the technique is task-based and does not lend itself to all possible goals and situations a user might encounter. How the data base or work-space is set up during an evaluation will also influence the complexity and resulting goals of the tasks.

We expected the cognitive walkthrough technique to do well in the area of action specification because of the questions on action-goal match and false-action match, and it did. Also, as mentioned above, some of the goal-related questions applied to the action level as well. CWs get at how well the USI object forms match a specific task goal, e.g., do the specific button label names match the meaning of the task goal, or is there a menu name so similar to the correct one that the user may be lead down the wrong path? Specific instances are evaluated with CW where a similar guideline would only say 'use clearly worded button labels', or 'use terminology familiar to the users'. One violated instance would result in guideline non-compliance but the same task-oriented evaluation of every USI object may or may not occur. CW actually evaluates the implementation of the advice provided in guidelines but mainly only for the action specification stage. With some work, this technique could be improved. To better assess the concept of semantic distance of execution, the technique would need to look at the number of steps and whether the user is likely to have knowledge of the correct sequence of steps for a particular user goal. More questions on the evaluation side of the cycle would need to be added. This technique has other shortcomings which have already been well documented in Wharton et al. (1992).

4.3 HEURISTIC EVALUATION

The heuristic method will always be largely dependent on the skill of the evaluator. In our case, the evaluators were fairly familiar with guidelines, and the traditional rules of good user interface design taught in USI design courses. Again, neither of these are heavily DM, graphical user interface oriented so it would not be expected that the evaluators would be familiar with or even think about concepts like semantic distance when doing an evaluation. The degree of familiarity the evaluators possess of the user's tasks would also play a large role in how well the evaluators performed against the stages model. Although other studies have shown that multiple evaluators increase the number of problems found, we did not find that to be the case here; rather, there was much overlap in the problems identified. The heuristic method appears to be a faster, less structured technique than guidelines and CW. The types of problems found overlapped quite a bit with those found by guidelines. Often the heuristic evaluators identified specific instances of a more general problem.

4.4 WEAKNESSES OF ALL THE TECHNIQUES

None of the evaluation techniques specifically made a distinction between whether the user-computer distances or mismatches are object or action oriented. For our particular application, many of the problems could not be easily classified in this manner. It seems like this distinction could provide important information, however, when evaluating issues such as feedback. The computer may provide the user with an indication that an operation was successful but information on which object the action was applied to may be lacking. Thus, if the user wished to delete a mission-request icon from the graphical scheduling display, and the selection and deletion actions were performed resulting in a feedback message "delete completed", the user would not be aware that an underlying mission icon was also inadvertantly selected and deleted. In this case, even a confirmation of deletion was not sufficient to prevent an error because the confirmation did not contain information about the objects to be deleted.

4.5 RECOMMENDATIONS AND SUMMARY

We have tried with this study to carry the assessment of USI evaluation techniques one step beyond the most recent work in this area. USI technology and implementation methods are growing and changing. As human factors professionals in the HCI field, we are responsible for understanding and evaluating the interaction between the computer interface design and the human's needs and goals. This study indicates that current evaluation methodologies are lacking

when it comes to assessing the DM, graphical-style interface for all stages or functional areas of HCI. Current evaluation techniques and training received by USI evaluators are still deeply influenced by the large amounts of research in the text-based, data entry style displays. There is a lack of understanding and guidance on assessing the intention formulation stage and the entire evaluation side of the HCI activity cycle. To begin assessing the concept of semantic distance for intention formation, techniques would need to assess:

- whether users are allowed to work at the level they are thinking,
- the number of actions to accomplish a single goal, and
- whether the user is likely to have knowledge of the correct sequence of actions for a single goal.

For the evaluation cycle, every icon, display object, and action needs to be evaluated as to whether it has meaning to the user and is at the level the user thinks. The feedback to every user response needs to be assessed as to whether the user can now determine whether their goal was met at every level. The concept of level is important to HCI and is addressed somewhat by Norman (1986). Users have many levels of intentions, and subsequent levels of action specification. When performing an evaluation, all of these levels need to be understood and the required input and output assessed accordingly. For a single task, there could be a task level intention, a series of sub-task intentions, each with a sequence of actions to accomplish each sub-task intention, and an individual action level. CW seemed to work well only at the evaluation of the single action level, neglecting the higher levels.

None of the techniques coherently addressed characteristics of the interface which classify it as a direct manipulation style interface. For example, characteristics of DM-style interfaces are (Schneiderman 1982 in Hutchins et al., 1986):

- are all actions rapid and reversible?
- is the input object also the output object?
- are there physical actions instead of complex syntax?

The system response time of the prototype was actually very slow. This seems to imply a failing in the general classification of the system as a DM system.

Work also needs to continue on models of the HCI process. The inability to easily compare results across studies which look at the effectiveness of different evaluation techniques points to a need for a general framework within which evaluation methodologies can be compared. We focused on one such frameworks here, the stages of user activity model, and touched on two others, Booth's variation model and the functional areas of Smith and Mosier. Each of these provided a slightly different view of the types of problems identified by different techniques. By using the framework suggested by the stages of user activity model to compare evaluation techniques, a more structured and cognitive-based approach to comparing evaluation techniques is possible. It too, however, could use some more detail, for example, in making the different levels more explicit. There appears to be many levels of evaluation which must occur to completely evaluate a system. A general or static evaluation (non-task based) can occur to answer questions such as those posed above on DM characteristics, for each system function. Guidelines can also be applied to assess the areas of data entry and sequence control. But a task-based evaluation also needs to occur to look at the sequencing and interrelationship of functions.

In the third phase of this research program, we plan to investigate whether it is possible to obtain evaluations of semantic distance and better assessments of the other stages from usability studies. A key to this type of evaluation is understanding user's goals and previous knowledge which implies a great need for user participation. We have high hopes that proper analysis techniques applied to usability study data can provide us with assessments on the directness of the user interface design.

SECTION 5

REFERENCES

Booth, P. A. (1990). ECM: A Scheme for Analysing User-System Errors. In D. Diaper et al. (eds.) *Human-Computer Interaction - INTERACT '90,* Elsevier Science Publishers, North-Holland, 1990, 47-54.

Hutchins, E. L., Hollan, J. D., and Norman, D. A. (1986). Direct Manipulation Interfaces. In D. A. Norman and S. W. Draper (eds.) *User Centered System Design: New Perspectives on Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. M. (1991). User Interface Evaluation in the Real World: A Comparison of Four Techniques. *CHI '91 Conference Proceedings,* 119-124.

Karat, C.M., Campbell, R., Fiegel, T. (1992). Comparison of Empirical Testing and Walkthrough Methods in User Interface Evaluation. *CHI '92 Proceedings*, 397-404.

Lewis, C., Polson, P., Wharton, C., & Rieman, J. (1990). Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces, *CHI '90 Proceedings*, 235-242.

Nielsen, J. and Molich, R. (1990). Heuristic Evaluation of User Interfaces. *CHI '90 Proceedings*, 249-256.

Norman, D. A. (1986). Cognitive Engineering. In D. A. Norman and S. W. Draper (eds.) *User Centered System Design: New Perspectives on Human-Computer Interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Olsen, J. R., and Olsen, G. M. (1990). The Growth of Cognitive Modelling in Human-Computer Interaction Since GOMS, *Human Computer Interaction*, 5, 221-265.

Rieman, J., Davies, S., Hair, D. C., Esemplare, M., Polson, P., and Lewis, C. (1990). An Automated Walkthrough: Description and Evaluation. Tech Rept. 90-18, Institute of Cognitive Science, University of Colorado, Boulder, CO.

Smith, S. L. and Mosier, J. N. (1986). Guidelines for Designing User Interface Software, ESD-TR-86-278, Bedford, MA: The MITRE Corporation.

Wharton, C., Bradford, J., Franzke, M., Jeffries, J. (1992). Applying Complex Cognitive Walkthroughs to more Complex User Interfaces: Experiences, Issues, and Recommendations, *CHI '92 Proceedings*, 381-388.

Weir, G. R. S. and Alty J. L. (1991). *Human-Computer Interaction and Complex Systems.* London: Harcourt Brace Jovanovich.

APPENDIX A

Problem Type Groupings

Data entry

Problem Type	Area	Prob. #	Method	Classification	Stage of user activity
Provide indication of acceptable data formats	Folder name Date format	33 11,71b, 60,63	H2 H1, 3Gl	Data entry	Action spec Art. dist.
Lack of data labels	Change layout-displayed SUA fields	30	H2 2 Gl	Data entry	Action spec Art. dist.
Poorly worded data labels	'Type in SUA'; 'Pick screen label'; Field labeled 'mission' but searching for request; Four report type labels are misleading; Two identically labeled time fields in create new mission	36 71 184 194 178	H2 Gl CW CW CW	Data entry Action goal match Action goal match False action match	Action spec Art. dist.
Entered data should be case insensitive	Folder names Mission labels	35	H2 Gl	Data entry	Action spec Art. dist. Execute
Poor grouping of data entry fields and data items	Reports time/date	41 102,86	H2 5 Gl	Data entry Data display	Action spec Interpret
Lack of or poorly placed data unit labels	Altitude time/date-reports date/time dial. Remarks label	72 41 42 65	2 Gl H2 H2 4 Gl	Data entry Data display	Action spec Interpret
Allow users to change or remove system default values	Global	84	Gl	Data entry	Action spec Execute
If defaults changed, revert to for rest of transactions	New mission changes std default	85	Gl	Data entry	Action spec Execute

Abbreviations not consistent and without a system; no prompting when abbrev. not recognized	Global	58 59	5 GI 1 GI	Data entry	Action spec Articulatory dist
No cues to indicate fixed or max. length of data	Global	66b	1 GI	Data entry	Action spec Articulatory dist
Differentiating between creating and editing folders	Folders	37,38	H2	Data entry	Action spec Articulatory distance
Not clear how to change from default time units to other units	Changing time from Z to EST	158	CW	Action-goal match Data entry	Action spec Articulatory distance
May try to edit data on the status-only indicators	Set date and time indicator; Changing mission request time	154 188	CW	False-action match Data entry	Action spec Articulatory distance
May forget to make data fields active by clicking on them before starting to type	Global	156	CW	Failure to add goal Data entry	Action spec Articulatory distance
May confuse the two identically labeled time fields	Create new mission	178	CW	False-action match Data entry	Action spec Articulatory distance
May think the time field on the find mission form applies to the start request time	Find mission	186	CW	Failure to add goals Data entry	Action spec Articulatory distance
Inconsistent data labels	Edit mission start date/time Find mission start date/time "pick", "select", "choose" Inconsistent fonts	89 89b 94	3 GI	Data entry Data display	Action spec/art. dist. Perception
Names/titles should be mandatory data	Mission icons Create folder	6 24,39	H1 H1, H2	Data entry	Action spec./art. dist.; Interpretation
May forget steps in sequence or how to use forms due to lack of prompts	Approve mission request; Find mission	161 182	CW	Failure to add goals Data entry	Action spec./art. dist.

No error msg when enter invalid data, lack of data validation	Del SUA name &edit Find mission Time Invalid SUA in folder	49 12 17 22 79,78	H2 H1 H1, H2 H2 3 Gl	Data entry User guidance	Evaluate/ Semantic dist.
Numbers too long, not formated	Mission # MAMS # SUA #	56 56a 56b 57	4 Gl	Data entry Data display	Execute Perception
Lack of input focus when windows first appear	Global	5	H1	Execute	Execute
No automatic justification of entered data	Global	75,75a, 68	3 GL	Data entry	Execute
Mission tapes difficult to select when timeline is large	Schedule	61	Gl	Data entry	Execute
Cursor not positioned usefully or consistently	Global	62	3 Gl	Data entry	Execute
Put mandatory fields first	Global	74	Gl	Data entry	Execute
Hard to fine tune the request icon position manually	Changing request time	189	CW	Hard to do Data entry	Execute
Lack of standard symbol for prompts	Global	66a,66c	2 Gl	Data entry User guidance	Interpret
May think they are in overstrike mode when are in insert mode (mode not obvious)	deleting default data	157	CW	Failure to add goal Data entry	Interpret; Articulatory distance
Lack of cues for row scanning	Global	76,97	2 Gl	Data entry	Perception
Lack of "undo" or way to reverse or backup to last input	Delete mission	3 3a, 3c, 121	H1 3 Gl	Data entry Seq control	Semantic distance/ Intention
Inability to apply an action to multiple objects at once	Mission icons Select SUAs Add/del SUAs Accept conflicts Edit missions	77 10 10a,32 20 27	Gl H1, H2 H2 H1 H1	Data entry	Semantic distance/ Intention

No indication of mandatory fields; May think need to fill in optional information/filters	Global; SUA and agency requests; Creating requests; Find mission	67 162 176 183	Gl CW CW CW	Data entry Extra goals	Semantic distance/ Intention
May not realize need to remove default data; Defaults may cause required fields not to be filled in with new data	Remove default SUAs from folder; Remove default data on create new mission; Time period of printed reports	168 177 192	CW	Failure to add goals Data entry	Semantic distance/ Intention

Data display

Problem Type	Area	Prob. #	Method	Classification	Stage of user activity
Order lists logically	Undisplayed SUAs	93	Gl	Data display	?
Blue color too saturated since not critical data	Schedule	107	Gl	Data display	Interpret/ Art. distance
Don't mix font case	Global	103	Gl	Data display	Perception
Mission names are unreadable when timeline is large	Schedule	2	H1	Data display	Perception
Difficult to see tapes in simultaneous mission	Schedule	47	H2 1 Gl	Data display Perception	Perception
When long time lines, grid lines overlap labels	Schedule	98,99	2 Gl	Data display	Perception
Provide/improve visual FB	Selected mission View button	18 28,129	H1 H1, H2 1 Gl	Data display Sequence control User guidance	Perception, Interpret/Art. distance, Semantic dist./Eval.
Inconsistent display formats and design standards	Global	87,88	2 Gl	Data display	Perception; Interpret/ art. dist.
Use blink coding for urgent items	Schedule	108	Gl	Data display	Perception; Interpret/ Art. distance

Sequence control

Problem Type	Area	Prob. #	Method	Classification	Stage of user activity
Selecting close button after create button not obvious; May skip create button and just invoke close buttion	Creating folder Removing SUAs from folder	169 170 174 175	CW	Action-goal match False-action match Sequence control	Action specification/ art. dist.
May think need to create request via button before they can create a mission with the create mission button.	Create new mission	179	CW	Adding extra goals Sequence control	Action specification/ art. dist.
Grey out non-active options	Scroll bars Deny option Deleting scheduled miss. Describe conflicts View mission menu	14 15 21 48 51	H1 H1 H1 H2 H2	Seq control	Action specification/ art. dist.
Button doesn't look like button	Day of wk button	44	H2	Sequence control	Action specification/ art. dist.
Not clear if filters for reports are 'and' or 'or'	Reports fields	29 193	H1, H2 CW	Sequence control Adding extra goals	Action specification/ art. dist.
Lack of confirmation for deletes	Del mission	3b, 50, 77b	2 GI H2	Sequence control User guidance	Action specification/ art. dist.
Two different actions occur on similar appearing data input fields	Folder name and Type in SUAs field	34	H2	Sequence control	Action specification/ art. dist.
SUA popup should not be a window	SUA	43	H2	Sequence control	Action specification/ art. dist.
Accept should close window	Describe confict	46	H2	Sequence Control	Action specification/ art. dist.
Lack of prompts, punctuation	Global	119,137 ,138	GI	Sequence Control User guidance	Action specification/ art. dist.

Control options available before mandatory info. entered	Open folder	9	H1,H 2	Seq control Action specification	Action specification/ art. dist.
No error msgs when select control options before mandatory info. entered	Create req.	1	H1	Seq control	Action specification/ interpret
Can't locate menu item	Set date and time under view menu; Folder option under Admin; Deny mission under schedule menu; Select reports under file menu;	153 166 180 190	CW	Action-goal match Sequence control	Action specification/ art. dist.
Think menu item is in a different menu	Approve mission request under schedule menu, not Mission menu; Reports	165 191	CW	False action match Sequence control	Action specification/ art. dist.
May think time bar controls date and time setting	Setting date and time	155	CW	False-action match Sequence control	Action specification/ art. dist.
May want to end dialogue box transaction with a return rather than or before selecting OK button	Global	159	CW	False-action match Sequence control	Action specification/ art. dist.
May not know to go to pending request to locate missions needing approval	Approve mission request	160	CW	Action-goal match Sequence control	Action specification/ art. dist.
Typing folder name to find folder is not obvious or consistent; May think can scroll through available SUA list	Selectingfolder	171 172	CW	Action-goal match False-action match Sequence control	Action specification/ art. dist.
May try to remove SUA by typing name in "Type SUA field"	Removing SUAs from folder	173	CW	False-action match Sequence control	Action specification/ art. dist.

May think View button allows them to view the found mission.	Find mission	185	CW	False-action match Sequence control	Action specification/ art. dist.
May select pending request to find mission W-555, but that is only for Phoenix Airspaces	Find mission	180	CW	False-action match Sequence control	Action specification/ art. dist.
Poorly worded/inconsistent button labels	Change screen, close,OK, cancel; Not obvious change screen button is required/right action; May select View in place of Change screen	26 45 114 163 187 164	H1 H2 3 Gl	Sequence control Action goal match	Articulatory dist./action specification
Put buttons in consistent locations	Global	54	H2 2 Gl	Sequence control	Execute
Arrows on scheduling scroll bar too small	Schedule	101	from Gl user	Sequence Control	Execute
Place cursor at most likely option in a list	Global	120	Gl	Sequence Control	Execute
Hierarchical menu may cause difficulty for selection	Folder option under Admin	167	CW	Hard to do Sequence control	Execute
Default system selection not indicated to user	Change screen when nothing selected	13	H1	Sequence control	Interpret/ art. dist.
Perform task analysis to identify related transactions	Global	113	Gl	Sequence Control	Outside of USI
Provide a shortcut for removing SUA from display	Display SUAs	53	H2	Sequence Control	Sem dist./ Intention
No FB for successful actions	Create folder	40	H2 Gl	Sequence Control	Semantic dist./ Eval.

User Guidance

Problem Type	Area	Prob. #	Method	Classification	Stage of user activity
Cursor should be readily distinguishable	Can't locate cursor, goes off screen	125	Gl	User guidance	Perception
Notify user when keyboard is locked	Locked out	128	Gl	User guidance	Execute
Error handling	Global	132,133 134, 136	4 Gl	User guidance	Interpret/ Evaluate
No dictionary provided of abbrev.s and codes	Global	143, 144	2 Gl	User guidance	Interpret
Provide display of past transactions	Global	145	Gl	User guidance	??
Error msg incorrectly worded	Describe conflicts "Tape" Passive rather than active	19, 89a 92, 115, 126, 131a, 47	H1 9 Gl	User guidance Interpret Sequence control	Interpret/ art. distance
Provide feedback while system is working, particularly when slow	Global	8, 8b, 130, 131	H1, H2 3 Gl	User guidance Interpret	Evaluate/ sem. dist
Slow system response time	Adding SUAs	32, 109	H2 6 Gl	User guidance Data display	??

APPENDIX B

TASK SCENARIO

For the following scenario you will be acting as a scheduler for the Phoenix Agency. The Phoenix Agency has a number of Special User Airspaces (SUAs) for which you will be responsible. These SUAs are: Canyon Run, Yankee 1, Yankee 2, India, W-556A, W556B, W556C, R-7221, R-7222 and R-7223 which is subdivided into R-7223N, R-7223S, R-7223E, and R-7223W. All of your airspaces are active or available for missions to be scheduled into them Monday through Friday from 0600 EST (1100 Z) to 1800 EST (2300 Z) except for India which is available 24 hours per day.

You have access to viewing and requesting SUAs in other agencies but you do not have authorization to schedule those airspaces.

1) You are planning a schedule for the week of 13-17 April 1992. All of the work done at Phoenix agency is done on EST. Set the screen start date and time appropriately.

2) Look at the requests for the airspaces you control, deny, or edit them as you deem appropriate. You cannot accept any conflicts.

3) Since you will be entering a number of missions that involve the same airspaces, create a folder named FIGHTWING that contains the following airspaces: Canvon Run, Yankee 1, Yankee 2, and India.

4) Create another folder named BOMBTEST that contains the following airspaces: R-7221, R-7222, and all the airspaces in R-7223.

5) Remove India from folder NIGHTRUN.

6) The attached requests have arrived by fax. Input them into the MAMS system as approved missions. If possible resolve any conflicts. You may do this by changing the start time of a mission, denying the mission, or changing the airspace if necessary. You may not accept any conflicts.

7) A squadron that does not have access to the MAMS system has asked you to check on their request called ASR on the 13 April 92 for W-555 in Neptune NAS. Has the request been scheduled, looked at (or not looked at), or denied? They also want to know about missions with the following MAMS numbers: 1230000 in R-8722W and 1280000 in W-554. Write the status of the mission on the back of this paper and set the paper aside to be faxed to the squadron.

8) Since ASR has been denied, the squadron has asked you to change the time of the request to 13 April 92 1300 EST.

9) You have been asked to change Bravo77 to a start time of 0900Z. Bravo77 has been scheduled daily over the next week in R-7223.

10) Print the following reports:
    - All missions for R-7222 and Canyon Run for the week of 13-17 April 1992.
    - All missions requested by Phoenix for the week of 13 April 1992.
    - Print Raider54 scheduled for 17 April 1992.